RPS92000036US1

PATENT

2700 AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Damon et al.
Serial No.: 09/675,545 Art Unit: 2126
Filed: September 28, 2000 Examiner: Diem K. Cao
For: AN EFFICIENT TIMER MANAGEMENT SYSTEM

RECEIVED
DEC 0 4 2003
Technology Center 2100

Mail Stop AF
C mmissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF (PATENT APPLICATION - 37 CFR 1.192)

1. Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on September 30, 2003.

NOTE: "The appellant shall, within 2 months from the date of the notice of appeal under § 1.191 in an application, reissue application, or patent under reexamination, or within the time allowed for response to the action appealed from, if such time is later, file a brief in triplicate." 37 CFR 1.192(a) (emphasis added).

2. STATUS OF APPLICANT

This application is on behalf of

☒ other than a small entity

☐ small entity

verified statement:

☐ attached

☐ already filed

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

☐ small entity $165.00

☒ other than a small entity $330.00

Appeal Brief fee due $330.00

CERTIFICATE OF MAILING (37 CFR § 1.8)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: 11/25/03

Serena Beller
(Type or print name of person mailing paper)

Serena Beller
(Signature of person mailing paper)

4. EXTENSION OF TERM

NOTE: The time periods set forth in 37 CFR 1.192(a) are subject to the provision of § 1.136 for patent applications. 37 CFR 1.191(d). Also see Notice of November 5, 1985 (1060 O.G. 27).

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply.

(complete (a) or (b) as applicable)

(a) ☐ Applicants petition for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

Extension (months)	Fee for other than small entity	Fee for small entity
☐ one month	$ 110.00	$ 55.00
☐ two months	$ 420.00	$ 210.00
☐ three months	$ 950.00	$ 475.00
☐ four months	$ 1,480.00	$ 740.00
Fee		

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

☐ An extension for _____ months has already been secured and the fee paid therefor of $_____ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request $_____

or

(b) ☒ Applicants believe that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicants have inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal Brief fee $330.00

Extension fee (if any) $0

TOTAL FEE DUE $330.00

6. FEE PAYMENT

☐ Attached is a check in the sum of $____

☒ Charge Account No. 50-0563 (RPS92000036US1) the sum of $330.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

☒ If any additional extension and/or fee is required, this is a request therefor and to charge Account No. 50-0563 (RPS92000036US1).

AND/OR

☒ If any additional fee for claims is required, charge Account No. 50-0563 (RPS92000036US1).

Reg. No.: 47,159

SIGNATURE OF ATTORNEY OR PATENT AGENT

Tel. No.: (512) 370-2832

Robert A. Voigt, Jr.
WINSTEAD SECHREST & MINICK P.C.
P.O. Box 50784
400 North Ervay Street
Dallas, Texas 75201

AUSTIN_1\237311\1
7036-P139US

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Damon et al.	:	Before the Examiner: Cao, Diem K.
Serial No. : 09/675,545	:	Group Art Unit: 2126
Filed: September 28, 2000	:	
	:	Intellectual Property Law
Title: AN EFFICIENT TIMER MANAGEMENT SYSTEM	:	IBM Corporation 972/B656
	:	P.O. Box 12195
	:	Research Triangle Park, NC 27709

APPEAL BRIEF

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on November 25, 2003.

Serena Beller
Signature

Serena Beller
(Printed name of person certifying)

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-34 are pending in the Application. Claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 stand rejected. Claims 6, 8-9, 11-14, 16-18, 22, 24-25, 27-30 and 32-34 stand objected to.

IV. STATUS OF AMENDMENTS

The Appellants' response to the Office Action, having a mailing date of April 10, 2003, has been considered, but the Examiner indicated that it did not place the Application in condition for allowance because the Appellants' arguments were deemed unpersuasive.

V. SUMMARY OF INVENTION

A timer is a device which can be set to furnish an interrupt or a timeout indication at a specific time instant or after a selected time interval. Specification, page 1, lines 10-11. Timers are required in data processing systems in which typical protocols require that a very large number of simultaneously occurring tasks or events be supervised to detect whether they occurred within predetermined delays. Specification, page 1, lines 11-13. For example, a start operation may be sent by the application to start the timer in order to supervise a corresponding event. Specification, page 1, lines 13-15. When the supervision of an event has to be interrupted for different reasons, a stop operation may be generated by the corresponding application. Specification, page 1, lines 15-16. After a while, the supervision of the corresponding event may be requested to start again, in which case a start operation may be generated by the application. Specification, page 1, lines 16-18. While the timer associated with an event is still running, the application may

request a restart operation in order to delay the timing of the corresponding event. Specification, page 1, lines 18-19.

A real-time operating system (RTOS), e.g., VxWorks™, OSE, Lynx, within a data processing system provides hardware resource management functions, e.g., memory management, task management, and data input/output management. Specification, page 2, lines 1-4. The operating system may further include a user interface for handling a screen display and the manipulation of a mouse. Specification, page 2, lines 4-5. The operating system may further include a module such as a timer management program, i.e., timer management system, for managing a plurality of timers that are started, stopped, idled, etc. by an application program in a data processing system. Specification, page 2, lines 5-7. Application programs, e.g., word processing software, database software, software for calculation for tables, reside on top of the OS in the topmost layer of a hierarchial software arrangement. Specification, page 2, lines 7-9.

Prior art timer management systems are used in either synchronous, i.e., single task, or asynchronous, i.e., multi-task, data processing systems. Specification, page 2, lines 10-11. In a synchronous system, when the timer expires, the application, i.e., user of the timer, is notified by a message commonly referred to as a timer message. Specification, page 2, lines 11-13. In an asynchronous system, when the timer expires, the timer message may be stored on queue before being sent to the application, i.e., user of the timer. Specification, page 2, lines 13-14. If the application stops the timer prior to receiving a timer message, the timer moves back to the idle state. Specification, page 2, lines 14-15. However, some operating systems do not allow timer messages to be removed from the queue when the application performs an operation on the expired timer prior to receiving the timer message. Specification, page 2, lines 15-17. Subsequently, a special state-variable in the timer message denotes the fact that it is an illegal time-out message. Specification, page 2, lines 17-18. In a synchronous system, this problem does not occur. Specification, page 2, lines 18-19.

It would therefore be desirable to develop a timer management system that is used in both a synchronous and asynchronous system where an asynchronous application is synchronously acting on the timer in an asynchronous system. Specification, page 2, lines 20-22. That is, it would therefore be desirable to develop a function to filter these illegal time-out messages that are transparent to the application. Specification, page 2, line 22 – page 3, line 1. It would further be desirable to allow a timer to be reinitialized without allocating new system memory. Specification, page 3, line 2.

The problems outlined above may at least in part be solved in some embodiments by a handle function that allows an asynchronous application in a multi-task system, i.e., asynchronous system, to synchronously act on the timer. Specification, page 4, lines 3-5. That is, when a timer in an asynchronous system times-out, the handle function filters the illegal time-out messages by allowing the asynchronous application to synchronously act on the timer. Specification, page 4, lines 5-7. In another embodiment of the present invention, a timer may be reinitialized from the same allocated block of memory used to create the timer. Specification, page 4, lines 7-8. In another embodiment of the present invention, a time-out message may be sent using the same allocated block of memory used to create the timer. Specification, page 4, lines 8-10.

In one embodiment, a timer management system for managing timers in both a synchronous and asynchronous system comprises an application program interface (API) for providing a set of synchronous functions allowing an application to functionally operate on the timer. Specification, page 4, lines 12-14. The timer management system further comprises a timer database for storing timer-related information. Specification, page 4, lines 14-15. Furthermore, the timer management system comprises a timer services for detecting the expiring of the timer. Specification, page 4, lines 16-17. A handle function of the timer services allows the asynchronous application, i.e., application in an asynchronous system, to synchronously act on the timer. Specification, page 4, lines 17-18. When the timer

expires, the handle function of the timer services allows the asynchronous application to act on the expired timer without incurring illegal time-out messages. Specification, page 4, lines 18-20.

VI. ISSUES

Are claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 properly rejected under 35 U.S.C. §103(a) as being unpatentable over Admitted Prior Art (APA) in view of Russell (U.S. Patent No. 6,349,388) and further in view of Dorn et al. (U.S. Patent No. 6,012,081) (hereinafter "Dorn")?

VII. GROUPING OF CLAIMS

Claims 1, 4, 7, 20 and 23 form a first group.

Claims 5 and 21 form a second group.

Claims 10, 15, 26 and 31 form a third group.

Claims 2, 3 and 19 should not be grouped together and should be considered separately.

The reasons for these groupings are set forth in Appellant's arguments in Section VIII.

VIII. ARGUMENT

A. The Examiner has not provided any motivation for combining APA, Russell and Dorn.

A *prima facie* showing of obviousness requires the Examiner to establish, *inter alia*, that the prior art references teach or suggest, either alone or in combination, all of the limitations of the claimed invention, and the Examiner must provide a motivation or suggestion to combine or modify the prior art reference to make the claimed inventions. M.P.E.P. § 2142. The showings must be clear and particular. *In re Lee*, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433-34 (Fed. Cir. 2002); *In re Kotzab*, 217 F.3d 1365, 1370, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000); *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). Broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence. *Id.*

In order to reject under 35 U.S.C. § 103, therefore, the Examiner must provide a proper motivation for combining or modifying the references. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1457-1458 (Fed. Cir. 1998); M.P.E.P. § 2142. The Examiner's motivation for modifying APA with Russell to have a timer database for storing timer-related information and a timer services detecting the expiring of a timer, as recited in claim 1 and similarly in claim 20, is to "provide a scalable approach to supporting an arbitrarily number of timers and reduces the typical processor overhead and hardware overhead involved in managing times (col. 2, lines 27-31)." Paper No. 8, page 3. The Examiner's motivation for modifying APA with Dorn to have a handle function that allows an application to act on an expired timer without incurring an illegal time-out message, as recited in claim 1 and similarly in claim 20, is to "provide a method to modify the behavior of an application." Paper No. 8, page 3. These are merely the Examiner's own opinions which does not amount to the required objective evidence to support a *prima facie* case of obviousness.

APA teaches a timer management program for managing a plurality of timers. Specification, page 2, lines 6-7.

Russell, on the other hand, teaches a timer processing engine that supports multiple virtual minimum time timers. Column 1, lines 50-65; Column 2, lines 27-32.

The Examiner must submit objective evidence and not rely on his own subjective opinion in support of combining a reference that teaches a timer management program for managing a plurality of timers with a reference that teaches a timer processing engine that supports multiple virtual minimum time timers. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Further, the Examiner must submit objective evidence and not rely on his own subjective for modifying APA to have a timer database for storing timer-related information and a timer services detecting the expiring of a timer. *Id.* Further, the Examiner must submit objective evidence and not rely on his own subjective for modifying APA to provide a scalable approach to

supporting an arbitrarily number of timers and reduces the typical processor overhead and hardware overhead involved in managing time. *Id.*

The Examiner's responds to Appellants' statements that the Examiner must submit objective evidence by asserting that when motivation comes from one of ordinary skill in the art, no objective evidence is needed. Paper No. 8, page 7. The Examiner is misguided. The Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). The Examiner must explain the reasons why one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious. *Id.*

Further, Dorn teaches threading control. Column 1, lines 5-7; Column 2, lines 50-57.

The Examiner must submit objective evidence and not rely on his own subjective opinion in support of combining a reference that teaches a timer management program for managing a plurality of timers with a reference that teaches threading control. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Further, the Examiner must submit objective evidence and not rely on his own subjective for modifying APA to have a handle function that allows an application to act on an expired timer without incurring an illegal time-out message. *Id.* Further, the Examiner must submit objective evidence and not rely on his own subjective for modifying APA to provide a method to modify the behavior of an application. *Id.*

The Examiner's responds to Appellants' statements that the Examiner must submit objective evidence by asserting that when motivation comes from one of ordinary skill in the art, no objective evidence is needed. Paper No. 8, page 7. The Examiner is misguided. The Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir.

2002). The Examiner must explain the reasons why one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious. *Id.*

Therefore, the Examiner has not presented a *prima facie* case of obviousness for rejection claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31.

B. APA, Russell and Dorn, taken singly or in combination, do not teach or suggest the following limitations.

Appellants respectfully assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein a handle function of said timer services allows said application to act on an expired timer without incurring an illegal time-out message" as recited in claim 1 and similarly in claim 20. The Examiner cites column 13, line 55 to column 15, line 12 of Dorn as teaching the above-cited claim limitation. Paper No. 5, page 3. Appellants respectfully traverse and assert that Dorn instead teaches a synchronous/asynchronous manager providing hooks to allow the application program to set and save thread specific data before and after a function invocation. Hooks may refer to instructions that provide breakpoints for future instructions. See www.techweb.com/encyclopedia. This language does not teach a handle function of the timer services where the handle function allows an application to act on an expired timer without incurring an illegal time-out message. Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said application performs the following operations on said timer via said API: creating said timer from an allocated block of said memory; activating said timer" as recited in claims 2-4. The Examiner cites column 8, lines 41-64, column 13, line 55 – column 15, line 12 of Dorn as teaching the above-identified limitations. Paper No. 8, pages 3-4. Appellants respectfully traverse and assert that Dorn instead teaches timer slots to handle

synchronous and asynchronous timers. This language is not the same as creating a timer. Instead, Dorn teaches slots to support timer controlled activities. Timer slots are not timers. Furthermore, Dorn does not teach creating a timer from an allocated block of memory. Furthermore, Dorn does not teach creating a timer from an allocated block of memory by an application via an API. Furthermore, this language does not teach activating a timer by an application via an API. Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "reinitiating said timer using said allocated block of system memory" as recited in claim 2. The Examiner asserts that one of ordinary skill in the art would modify APA with Dorn where Dorn teaches that the time slots are reusable in order for APA to reinitiate a timer using an allocated block of system memory. Paper No. 8, page 10. The Examiner has provided no evidence that the time slots in Dorn are reusable. Further, the Examiner has provided no evidence that a reusable time slot corresponds to reinitiating a timer using an allocated block of system memory. The Examiner is merely relying upon his own subjective opinion. Further, the Examiner states that the motivation to modify APA with Dorn can come from one of ordinary skill in the art and that no objective evidence is needed. Paper No. 8, page 10. As stated above, the Examiner is misguided. The Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). The Examiner must explain the reasons why one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious. *Id.* Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said timer expires and said timer services sends synchronously a time-out message to said application, wherein said time-out message is sent using said allocated block of system memory" as recited in claim 3. The Examiner states:

> Dorn does not explicitly teach the time-out message is sent using the allocated block of system memory. Dorn teaches the time-out message is sent to an internal queue (the expiration event...internally; col. 14, lines 35-39). It would have been obvious to apply the teaching of Dorn to the system of APA because it provides the programmers not to bother with the low level details. Paper No. 8, page 4.

Instead, Dorn teaches that if an interval timer expires before the invocation of the callback function, the expiration events will be queued internally. Column 14, lines 35-39. This language does not teach a time-out message. Further, this language does not teach that a time-out message is sent using an allocated block of system memory. Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

The Examiner further states that objective evidence is not needed to modify APA with Dorn when the motivation comes from one of ordinary skill in the art. Paper No. 8, pages 10-11. As stated above, Appellants respectfully traverse that assertion and assert that the Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said particular queue is a system queue attached to said application" as recited in claims 5 and 21. The Examiner cites column 14, lines 35-39 of Dorn as teaching the above-identified limitation. Paper No. 8, page 5. Instead, as stated above, Dorn teaches that if an interval timer expires

before the invocation of the callback function, the expiration events will be queued internally. This language does not teach a system queue attached to an application. Further, the Examiner states that "it would have been obvious to modify the queue in the system of Dorn to attach to the application in the system of APA because it serves the same purpose." Paper No. 8, page 5. This is not a proper motivation for modifying Dorn. The Examiner must submit objective evidence and not rely on his own subjective opinion in support of combining APA, Russell and Dorn. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said timer is activated by said application, wherein said timer is in a running state in said asynchronous state machine with said time-out message being queued" as recited in claim 10 and similarly in claims 15, 26 and 31. The Examiner states that one of ordinary skill in the art would be able to modify APA to have an application activate the timer where the timer is in a running state with the time-out message being queued. Paper No. 8, page 12. The Examiner further states that no objective evidence is required to support the motivation to modify APA. Paper No. 8, page 12. As stated above, the Examiner is misguided. The Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). The Examiner must explain the reasons why one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious. *Id.* Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

Appellants further assert that APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said API is a DLL file" as recited in claim 19. The Examiner states that one of ordinary skill in the art would have been motivated to modify APA to have an API be a DLL file. Paper No. 8, page 12. The Examiner further states that no objective evidence is required to support the motivation to modify APA when the motivation comes from one of ordinary skill in the art. Paper No. 8, page 12. As stated above, the Examiner is misguided. The Examiner must submit objective evidence in support of combining references whether the motivation comes from one of ordinary skill in the art or in the references themselves. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). The Examiner must explain the reasons why one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious. *Id.* Therefore, the Examiner has not presented a *prima facie* case of obviousness, since the Examiner is relying upon an incorrect factual predicate in support of the rejection. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1455 (Fed. Cir. 1998).

As a result of the foregoing, Appellants respectfully assert that there are numerous claim limitations not taught or suggested in the cited prior art, and thus the Examiner has not presented a *prima facie* case of obviousness for rejecting claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 in view of the cited prior art.

IX. CONCLUSION

For the reasons noted above, the rejections of claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 are in error. Appellant respectfully requests reversal of the rejections and allowance of claims 1-34.

Respectfully submitted,
WINSTEAD SECHREST & MINICK P.C.
Attorneys for Appellants

By: _____
Robert A. Voigt, Jr.
Reg. No. 47,159
Kelly K. Kordzik
Reg. No. 36,571

P.O. Box 50784
400 North Ervay Street
Dallas, Texas 75201
(512) 370-2832

APPENDIX

1. A timer management system for managing timers in both a synchronous and asynchronous system comprising:

an application program interface (API) providing a set of synchronous functions allowing an application to functionally operate a timer;

a timer database for storing timer-related information; and

a timer services detecting the expiring of said timer, wherein a handle function of said timer services allows said application to act on an expired timer without incurring an illegal time-out message.

2. The timer management system as recited in claim 1, wherein said application performs the following operations on said timer via said API:

creating said timer from an allocated block of system memory;

activating said timer; and

reinitializing said timer using said allocated block of system memory.

3. The timer management system as recited in claim 1, wherein said application performs the following operation on said timer via said API:

creating said timer from an allocated block of system memory; and

activating said timer;

wherein said timer expires and said timer services sends synchronously a time-out message to said application, wherein said time-out message is sent using said allocated block of system memory.

4. The timer management system as recited in claim 1, wherein said application performs the following operation on said timer via said API:

creating said timer from an allocated block of system memory; and

activating said timer;

wherein said timer expires and said timer services sends a time-out message to a particular queue, wherein said timer is in an expired state in an asynchronous state machine.

5. The timer management system as recited in claim 4, wherein said particular queue is a system queue attached to said application.

6. The timer management system as recited in claim 4, wherein said application receives said time-out message, wherein said handle function transfers said timer from said expired state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

7. The timer management system as recited in claim 4, wherein said application stops said timer, wherein said timer is in an idle state in said asynchronous state machine with said time-out message being queued.

8. The timer management system as recited in claim 7, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

9. The timer management system as recited in claim 7, wherein said timer is deleted by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state

machine, wherein said handle function allows said application to synchronously act on said timer.

10. The timer management system as recited in claim 7, wherein said timer is activated by said application, wherein said timer is in a running state in said asynchronous state machine with said time-out message being queued.

11. The timer management system as recited in claim 10, wherein said timer is deleted by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

12. The timer management system as recited in claim 10, wherein said timer is stopped by said application, wherein said timer is in said idle state in said asynchronous state machine with said time-out message being queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

13. The timer management system as recited in claim 10, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said running state in said asynchronous state machine to a running state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

14. The timer management system as recited in claim 4, wherein said application deletes said timer, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

15. The timer management system as recited in claim 4, wherein said application activates said timer, wherein said timer is in a running state in said asynchronous state machine with said time-out message being queued.

16. The timer management system as recited in claim 15, wherein said timer is deleted by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

17. The timer management system as recited in claim 15, wherein said timer is stopped by said application, wherein said timer is in an idle state in said asynchronous state machine with said time-out message being queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

18. The timer management system as recited in claim 15, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said running state in said asynchronous state machine to a running state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

19. The timer management system as recited in claim 1, wherein said API is a DLL file.

20. A method for managing timers in both a synchronous and asynchronous system comprising the steps of:

creating a timer from an allocated block of system memory by an application via an application program interface (API);

activating said timer;

expiring of said timer; and

sending a time-out message to a particular queue when said timer expires, wherein said timer is in an expired state in an asynchronous state machine, wherein a handle function allows said application to act on said expired timer without incurring an illegal time-out message.

21. The method as recited in claim 20, wherein said particular queue is a system queue attached to said application.

22. The method as recited in claim 20 further comprising the step of:

receiving said time-out message by said application, wherein said handle function transfers said timer from said expired state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

23. The method as recited in claim 20 further comprising the step of:

stopping said timer by said application, wherein said timer is in an idle state in said asynchronous state machine with said time-out message being queued.

24. The method as recited in claim 23, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

25. The method as recited in claim 23 further comprising the step of:

deleting said timer by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

26. The method as recited in claim 23 further comprising the step of:

activating said timer by said application, wherein said timer is in a running state in said asynchronous state machine with said time-out message being queued.

27. The method as recited in claim 26 further comprising the step of:

deleting said timer by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a

synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

28. The method as recited in claim 26 further comprising the step of:

stopping said timer by said application, wherein said timer is in said idle state in said asynchronous state machine with said time-out message being queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

29. The method as recited in claim 26, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said running state in said asynchronous state machine to a running state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

30. The method as recited in claim 20 further comprising the step of:

deleting said timer by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

31. The method as recited in claim 20 further comprising the step of:

activating said timer by said application, wherein said timer is in a running state in said asynchronous state machine with said time-out message being queued.

32. The method as recited in claim 31 further comprising the step of:

deleting said timer by said application, wherein said timer is in a state in said asynchronous state machine in which said timer is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said state in said asynchronous state machine in which said timer is deleted and said time-out message is queued to a non-existent state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

33. The method as recited in claim 31 further comprising the step of:

stopping said timer by said application, wherein said timer is in an idle state in said asynchronous state machine with said time-out message being queued, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous state machine to an idle state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.

34. The method as recited in claim 31, wherein said time-out message is dequeued, wherein said handle function transfers said timer from said running state in said asynchronous state machine to a running state in a synchronous state machine, wherein said handle function allows said application to synchronously act on said timer.